
Subject: Re: About DHCtrl and window handles...
Posted by [mdelfede](#) on Mon, 12 Nov 2007 20:53:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Mon, 12 November 2007 20:39mdelfede wrote on Mon, 12 November 2007 07:02luzr wrote on Mon, 12 November 2007 12:23
Well, the trouble is that this is (and has to be) platform specific.

Of course... but there is a thing that I don't like in Upp, and is that many platform specific methods are made public... I'd separate them from the public interface.

What is that supposed to solve?

Well, that should force derived controls to be platform-independent, if methods are made private, or at least that would force user code (not derived from Ctrl) to **not** use platform dependent stuffs.... IMHO those should be good reasons.
IMHO, core classes should encapsulate all platform dependent behaviour.

Quote:

BTW, there are people suggest to me to make ALL methods public and virtual...

Brrrrrrr !

So, if they don't bother of writing portable code, here's again the way :

```
#define private public  
#include <all-Upp>
```

That's just for public, for 'virtual' I guess it's not so easy!

Quote:

Quote:

Ok, but I don't understand if EventProc is called on each control for all Windows or not.

Normally, only to top-level Ctrl's (GetParent() == NULL).

Of course, existing DHCtrl receives them too. So better to say, to all windows with handles, but handles for non-top-level windows are yet to be implemented in X11, that is the important part of the task...

So, as I'm trying to implement it (to some extents...), what I need to know is what happens now if I create an X11 handle on a child control, in respect with events.
I suppose X11 sends the event directly to window inside my control... so, my code **or** Upp has to have a way to handle the event, what is done usually in an event loop.

As you're keeping a list of X11 handles, I suppose that you've got a global event handler that dispatches in some way events to controls, right ? Because that's the central point... if Upp dispatches in some way the event, I must just catch it; if not, I must hook somehow my code in your event loop... If I'm not making some big mistake, of course

Quote:

Quote:

- Both A and B once, with A as parameter
- Only to A, as B don't need repaint

EventProc is already redirected to the window with given handle, if that answers your question....

Not completely... I understand that your code redirects events to correct control, but what I don't know is if your code catches ALL events or only mainwindows events.

Better explained, if I create an X11 window that overlap a control, you're saying that my X11 window gets the event directly ? does UPP receive it also and dispatch it ?

Is maybe that one the purpose of having a global list of X11 windows handles ?

.....

Quote:

Quote:

And to windowed ones ?

There is a map in X11 platform specific code that maps handles to Ctrl *.

Ok, so I guess I did understand right... it's the Xwindow()[] map, right ? So, if I hook my handle inside it, my control gets Events as usual non-windowed controls ?

Quote:

Quote:

Again an example :

A is a top level window, and B is a child of A, but with window handle too. What happens if I click on B ?

- B gets the event directly, A gets nothing

In Win32 DHCtrl, this is the case.

So, I guess I'll have to duplicate this behaviour.

Quote:

Quote:

And, last question (sorry, but it's a complicated matter...), another example :

-A is a main window (with XWindow inside), B is a child of A (no matter if windowed or not...) and C is a child of B.

Hey, wait a moment. This is common misconception. Parentship is not the same thing as ownership.

All top-level widgets have no parents. Anyway, they can have owners.

Yes, I know the difference....

A child is usually an included control (or at least a dependent control), ownership is related to create/destroy it, not to the appearance nor to event handling.

Well, I'll try to go a bit further with my X11_DHCtrl

Ciao

Max

ops.... just another question :

Suppose this one :

MainWindow->A->B (-> represent Parent of)

B is windowed. I get the event... should I reflect to toplevel window ? I think that should be the more consistent way to do the job...

Ciao

Max
